

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S)			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School		6b. OFFICE SYMBOL (If applicable) 37		7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School	
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000			7b. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION		8b. OFFICE SYMBOL (If applicable)		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code)			10. SOURCE OF FUNDING NUMBERS		
			Program Element No	Project No	Task No
					Work Unit Accession Number
11. TITLE (Include Security Classification) ANCHORING-AND-ADJUSTMENT IN SOFTWARE PROJECT MANAGEMENT: AN EXPERIMENTAL INVESTIGATION					
12. PERSONAL AUTHOR(S) Timothy P. Costello					
13a. TYPE OF REPORT Master's Thesis		13b. TIME COVERED From To		14. DATE OF REPORT (year, month, day) 1992, September	
				15. PAGE COUNT 77	
16. SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.					
17. COSATI CODES			18. SUBJECT TERMS (continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUBGROUP	Software Project Management; Cognitive Heuristics; Biases; Dynamic Decision Making; Anchoring-and-Adjustment		
19. ABSTRACT (continue on reverse if necessary and identify by block number)					
<p>Software project development continues to be characterized by cost overruns, late deliveries, poor reliability and user dissatisfaction. The Systems Dynamics Model of Software Project Management is a quantitative model of software project dynamics that is attempting to gain some valuable insight into the managerial side of developing software systems.</p> <p>The objective of this thesis was to use the Systems Dynamics Model's gaming interface to investigate the cognitive heuristic anchoring-and-adjustment in dynamic decision environments, and its use in software project management. Specifically, subjects were provided with either a low or a high anchor condition to determine the effect on subject productivity estimation and project performance when confronted with dynamic decision making in software project management. The results show that subjects used anchoring to simplify decision making in the complex dynamic environment. There was evidence of bias introduced by the anchor, thereby causing dysfunctional performance.</p>					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS REPORT <input type="checkbox"/> DTIC USERS				21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Professor Kishore Sengupta				22b. TELEPHONE (Include Area code) (408)646-3212	
				22c. OFFICE SYMBOL AS/SE	

Approved for public release; distribution is unlimited.

Anchoring-and-Adjustment in Software Project Management:
An Experimental Investigation

by

Timothy P. Costello
Lieutenant, United States Navy
B.S., United States Naval Academy, 1985

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
September 1992

ABSTRACT

Software project development continues to be characterized by cost overruns, late deliveries, poor reliability and user dissatisfaction. The Systems Dynamics Model of Software Project Management is a quantitative model of software project dynamics that is attempting to gain some valuable insight into the managerial side of developing software systems.

The objective of this thesis was to use the Systems Dynamics Model's gaming interface to investigate the cognitive heuristic anchoring-and-adjustment in dynamic decision environments, and its use in software project management. Specifically, subjects were provided with either a low or a high anchor condition to determine the effect on subject productivity estimation and project performance when confronted with dynamic decision making in software project management. The results show that subjects used anchoring to simplify decision making in the complex dynamic environment. There was evidence of bias introduced by the anchor, thereby causing dysfunctional performance.

1 K0513
C 756144
c.1

TABLE OF CONTENTS

I.	INTRODUCTION	1
A.	BACKGROUND	1
B.	PREMISE OF RESEARCH	4
	1. Dynamic Decision Making	4
	2. Anchoring-and-Adjustment	5
	3. Software Project Estimations	7
C.	RESEARCH HYPOTHESES	10
D.	EXPERIMENTAL APPROACH	10
E.	THESIS ORGANIZATION	12
II.	METHOD	13
A.	EXPERIMENTAL DESIGN	13
B.	TASK ENVIRONMENT	14
C.	EXPERIMENTAL SUBJECTS	16
D.	EXPERIMENTAL SETTING	19
	1. Software	19
	2. Reports Provided at each Time Interval	20
	3. Information Provided to Subjects	22
E.	DEPENDENT MEASURES	25
III.	EXPERIMENT RESULTS AND ANALYSIS	26
A.	MAIN EXPERIMENT	26

1. Subjects' Productivity Estimates	27
a. Between-Subjects Effects	29
(1) Different Anchors Effect	29
(2) Different Projects Effect	29
b. Within Subjects Effects	30
(1) Time Effect	30
(2) Time and Different Anchors Effect	30
(3) Time and Different Projects Effect	30
2. Subjects' Performance	31
a. Different Anchors Effect	33
b. Different Order Effect	34
c. Anchor in Different Order Effect	34
B. COGNITIVE MODELS	34
IV. CONCLUSIONS	40
A. SUMMARY OF OBJECTIVES	40
B. SUMMARY OF RESULTS	41
1. H_1 : Anchoring-and-Adjustment is Used	41
2. H_2 : Performance is Affected	41
C. IMPLICATIONS OF RESULTS	42
D. LIMITATIONS AND FUTURE RESEARCH	43
APPENDIX A	46
APPENDIX B	47

APPENDIX C	57
APPENDIX D	64
LIST OF REFERENCES	68
INITIAL DISTRIBUTION LIST	70

I. INTRODUCTION

A. BACKGROUND

In today's information based society the demand for complex computer software to run on constantly improving hardware is far greater than the industry's ability to produce it. Computer hardware performance has increased a thousand-fold in the last 30 years while improvements in software development have been anemic by comparison. Hardware costs are declining, customer demand is high, the number of end users is increasing, and programming productivity is essentially flat (Moore, 1982). If the current trends in software supply and demand are projected out to the year 2040, the entire population of the United States would have to be software programmers in order to satisfy the demand (Kitfield, 1989).

Software development continues to be characterized by cost overruns, late deliveries, poor reliability and end user dissatisfaction. As the complexity of software continues to rise, so do the ambiguity of schedules, budgets and performance criteria. Even with the introduction of modern software engineering techniques, software development continues to be a creative process, highly dependent upon programmer ability, experience, and intuition. Although there is a significant

amount of literature available describing software complexity and the effect of programmer capability on software productivity, too little attention has been given to the effects that project management has on software development rates.

Managing software development is a complex process of controlling interrelated abstract entities (e.g., personnel turnover, requirements changes, staff productivity, project complexity, budgets, etc.) in a dynamic environment. The project manager must continuously assess the status of this environment to make reliable estimations and cognizant decisions. Each estimation and subsequent decision the manager makes has a dynamic effect on the entire system (Abdel-Hamid, 1989).

The causal processes faced by software project managers contain feedback loops, time delays, and non-linearities, all of which severely inhibit effective forecasting and decision making. Over a project's lifecycle, managers are presented with volumes of unreliable and even conflicting software metrics data to base their decisions on. Under these conditions, software managers are faced with the highly ambiguous task of controlling the development process.

How can software project managers hope to be effective, when the management process itself is so ambiguous? A better understanding of how software managers cope (or are unable to cope) in such a complex environment is needed before

significant improvements in software development performance can be realized.

The Systems Dynamics Model (SDM) of Software Project Management is a quantitative model of software project dynamics that has attempted to gain some valuable insight into the managerial side of developing software systems (Abdel-Hamid and Madnick, 1988). It is a comprehensive simulation model of the software development process that integrates both the management type functions (e.g., planning, controlling and staffing) with the software production type activities (e.g., design, coding, reviewing and testing).

The SDM's gaming interface enables users to directly interact with the simulation model. Variables can be displayed, reports can be generated, and calculations can be made to provide the user with a complete simulation of the management environment. Users can also influence the environment by making estimations and dynamic decisions regarding management variables.

Through the use of the SDM and its gaming interface, a wide range of managerial processes and complex operating environments can be simulated, tested and evaluated. The gaming interface of the Systems Dynamics Model provides an effective means of studying the dynamic decision making process software project managers experience in real world environments.

B. PREMISE OF RESEARCH

1. Dynamic Decision Making

Dynamic decision making is a continuous process of making decisions in an environment being conditioned by prior decisions. Each decision not only alters the environment, but alters reference points used to make future decisions (Paich and Sterman, 1992).

Dynamic decision making is performed everyday in simple settings. When the causal process is fully understood by the decision maker, dynamic decision making can lead to success. For example, an experienced artist trying to make a certain color starts with a base color and systematically adds different colors to make the desired color. Each time the artist chooses a color to add, a dynamic decision has been made. The added color changes the base color and effects the artist's next choice. The dynamic decision making continues until the desired color is made.

But what if the causal process is not completely understood by the decision maker? If the artist in the above example only had a "best guess" as to which colors to use, would the small imperceivable mistakes present in each estimation lead the dynamic decision making process to eventual success, or failure?

Software project management is an example of dynamic decision making in a complex environment. As stated earlier,

the casual processes in software project management are complex and not easily understood. How then, do software project managers form the estimations used in the dynamic decision making process leading them to eventual success, or failure?

2. Anchoring-and-Adjustment

Prior work in dynamic decision making has shown that the mental models people use to manipulate dynamic environments are usually inadequate (Paich and Sterman, 1992). As projects become larger and more complex, managers tend to rely increasingly on simple cognitive heuristics to make decisions (Tversky and Kahneman, 1974).

One of the simple cognitive heuristics managers use to simplify complex environments is called "anchoring-and-adjustment". Anchoring is a behavioral phenomenon where a given variables' heuristic is unduly relied upon in making future adjustments to the variable (Tversky and Kahneman, 1974). In other words, when asked to make an estimation, different starting points yield different estimates, because the estimates are unduly biased toward the initial starting point. Instead of making estimations based purely on environmental factors, "anchoring-and-adjustment" is used to simplify the decision making process used to formulate the estimate. The use of such simple judgmental operations can result in

cognitive biases leading to dysfunctional performance (Tversky and Kahneman, 1974).

However, Hogarth (1981) has argued that past demonstrations of this decisional bias with dysfunctional performance may have been a product of the discrete and static nature of the tasks and environment tested. Hogarth (1981) goes on to state:

...in continuous environments, the adjustment and anchoring heuristic essentially provides the basic mode of judgment. Consider, for instance, how one forms impressions of strangers through interaction. That is, in discrete incidents a single (possibly inaccurate) judgement is made. In continuous processing, however, a series of adjustment and anchoring responses, all of which may be relatively inaccurate, takes one progressively to the target. (p. 206)

According to Hogarth (1981), studies of decisional behavior should be performed in dynamic environments where feedback is allowed to play a role in the judgmental process, "...theories of judgment and choice that lack a continuous perspective exclude one of the most important determinants of the behavior they purport to explain." (p. 213) In a dynamic environment the dysfunctional bias introduced by the adjustment-and-anchoring heuristic would have a reduced effect on overall performance, and in fact, is a normal entity in the judgmental process eventually leading to success.

Hogarth (1981) described this process as the probability of hitting a fixed target in a dynamic environment. Imagine a marksman trying to hit a target some distance away. After each shot, the marksman is allowed to take a step closer

to the target, thus improving the probability of hitting the target with each step. In effect, the marksman was anchored to his initial position and then adjusted positions progressively closer to the target based on feedback available in the dynamic environment.

Now try to imagine the results of using the same anchoring-and-adjustment technique, if after each time the marksman takes a step, the target was somehow influenced by the last shot, changing its position. The marksman may, or may not have moved closer to the target. This dynamic decision making environment is now analogous to software project management, where prior estimations affect the position of the target after receiving feedback making it more difficult to hit.

3. Software Project Estimations

An example of one of the many moving targets a software project manager must try to hit is the project schedule. Figure 1-1 is a causal loop diagram that represents just one of the loops showing how project estimates influence the position of the 'target' schedule, making it difficult to hit.

Project estimates of productivity indirectly affect the work force hiring and firing decisions by influencing the estimated schedule. Inaccurate estimates can have a severe impact on the entire system because of the relationship

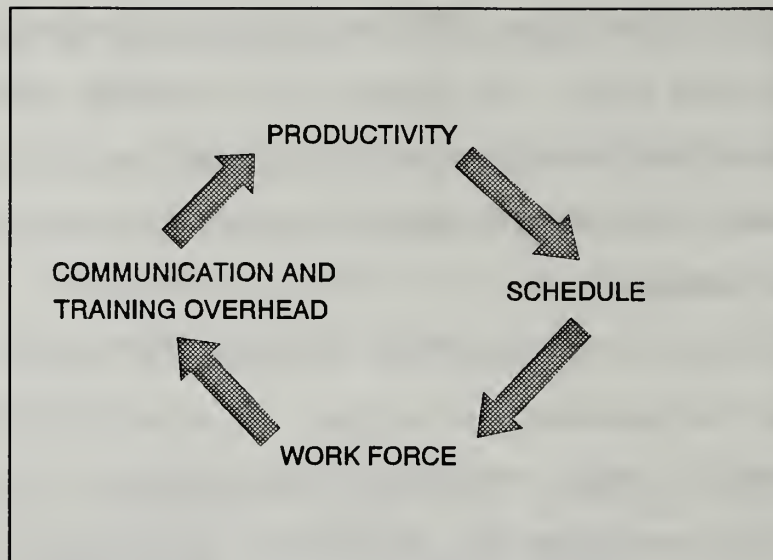


Figure 1-1 Causal Loop Diagram

between staff size, communication and training overhead, and productivity (Abdel-Hamid, 1988).

If productivity estimates are too high, the perceived staff size needed will be lower. Decreasing the staff size reduces communication and training overhead which in turn increases their productivity. This moves the actual productivity towards the inflated estimate of productivity until the increased pressure put on the undermanned staff causes an increase in the turnover rate.

If productivity estimates are too low, the perceived staff size needed will be higher. Increasing the staff size expands the communication and training overhead which in turn decreases their productivity. This moves the actual productivity towards the depressed estimate of productivity until

management realizes that more time is being spent on communication and training overhead than on the project itself.

The validity of project estimates therefore have a strong influence on the estimated schedule, hiring and firing decisions, communication and training overhead, and productivity (Abdel-Hamid, 1988).

Several studies have been performed exploring the "anchoring-and-adjustment" heuristic in laboratory and information rich, "real world" environments (Tversky and Kahneman, 1986; Paich and Sterman, 1992). However, a vast majority of them have been conducted in static environments. The phenomenon of anchoring-and-adjustment in dynamic environments has been examined in very few studies. For example, Ronan (1990) conducted an experiment regarding anchoring-and-adjustment in a dynamic environment, just as Hogarth suggested. The experiment concluded that subjects acting as software project managers did indeed rely on the "anchor" to reduce the complexity of making productivity estimations to a simpler judgmental operation. However, the subjects' estimates were not actually used in the model. Therefore the 'target' was not affected by the subjects' estimates and did not move over the project's lifecycle.

C. RESEARCH HYPOTHESES

The discussion presented thus far has suggested some interesting questions left unanswered, thereby suggesting possible conjectures and hypotheses. This thesis investigated the anchoring-and-adjustment phenomenon in a dynamic software development environment where dynamic decision making by project management was used to control the project. Two hypotheses were tested:

H_1 : When given a task of making staff productivity estimations in a dynamic environment, different anchors operationalized as initial estimates on the same project will produce different estimations on a continuing basis.

H_2 : When given a task of making staff productivity estimations in a dynamic environment, different initial estimates on the same project will lead to different performance results.

D. EXPERIMENTAL APPROACH

The objective of this thesis was to design, construct and execute an experiment, using an enhanced version of the SDM gaming interface, to investigate software project management heuristics involving "anchoring-and-adjustment" and the role it plays in dynamic environments involving dynamic decision making. The experiment employed a within-subjects experimental design, wherein subjects ran two separate software project

simulations in order to expose them to both a low and a high anchor testing environment.

Average staff productivity was chosen as the project management variable subjects would be estimating because of its relative importance to a project manager's ability to effectively manage a project to a successful and timely completion. The estimate of the staff's average productivity directly impacts on the staff's size as was described by Figure 1-1, and can have a major effect on total project duration and cost.

The SDM gaming interface was altered to present each subject with a standard interface to the simulation model. Each subject was exposed to a productivity "anchor" at the beginning of a project and then required to make productivity estimates (in Tasks/man-day) through the integration and testing phases of a project's development. Before each estimation, the subjects were given feedback in the form of reports provided by the simulation's project staff (played by the SDM). The subjects' goal for the simulation was to provide the most accurate estimation of the staff's overall average productivity so that the project could be completed within an established number of work-days.

The majority of research on decision making has focused on data which reflect only the end product of the decision process (Payne, 1976). So a second research procedure was employed by having a small group of subjects verbally recorded

their thoughts while performing each simulation. The transcripts produced were *protocols* of their decision-making behavior (Bouwman, 1983). A simple protocol analysis translating the transcripts into a more accessible representation was made to augment the empirical data from the main experiment.

The subjects in this experiment were fifth-quarter graduate students (in a six-quarter curriculum) studying in the Computer Systems Management curriculum at the Naval Postgraduate School.

E. THESIS ORGANIZATION

Chapter II describes the methodology used for the design and execution of the experiment. Chapter III states the experimental results. Chapter IV summarizes the findings of the experiment and describes its implications.

II. METHOD

A. EXPERIMENTAL DESIGN

Table 2.1 represents the experimental design of the thesis. The experiment employed a within-subjects experimental design, wherein subjects ran two separate software project simulations so as to expose them to both a low and a high anchor environment. Accordingly, the experiment was divided into four separate groupings.

Table 2.1 EXPERIMENTAL DESIGN

Group #	First Simulation	Second Simulation
Group 1	Project 1 Low Anchor	Project 2 High Anchor
Group 2	Project 1 High Anchor	Project 2 Low Anchor
Group 3	Project 2 Low Anchor	Project 1 High Anchor
Group 4	Project 2 High Anchor	Project 1 Low Anchor

One subject from each group was given a tape recorder to record his or her thoughts for a simple protocol analysis of the decision processes involved.

For each simulation, final project duration, the subject's input for average staff productivity and the effect it had on the project each time interval, were recorded.

B. TASK ENVIRONMENT

The basic task the subjects were asked to perform was set up to be similar in many ways to the flight simulators that pilots use to mimic flying an aircraft from takeoff at point A to landing at point B. Instead of flying an aircraft, the SDM gaming interface mimics the life of a real software project from the start of the "implementation" phase to the end of the "testing" phase. Instead of being an aircraft pilot, the test subject played the role of a valuable assistant to a software project manager. In less than an hour the subject lived through a project's life-cycle as an active participant in its management.

Specifically, their role was to track a software project's progress using a number of reports produced for them every 40 work-days. After each 40 work-day time interval, they were required to submit their best estimate of the project staff's overall average productivity (in Tasks/man-day). Their estimate was then used by the simulation's project manager (played by the SDM) to make the necessary adjustments to the project's staff size in order to complete the project on schedule with the least amount of resources. This cycle, of report generation by the model and estimated average productivity input from the subject, then continue until the project was completed. The subject's goal for the exercise was to

ensure the project was completed within the allotted schedule duration (given in Work-days) with the least amount of resources.

By giving the subject a forecast of the overall average staff productivity expected for the project, an anchor was introduced. The subject then made his or her own estimation of the team's average productivity based on the reports generated by the staff each time interval and the forecasted anchor given by management from the start. Any bias towards the anchor, and the decision processes involved during each time interval, were then recorded, measured, and analyzed.

Two separate and distinct software projects were selected to be used in the experiment. By using real projects with real data, the results of the experiment can be measured, compared and validated against a known baseline.

Project #1 was a real software project developed in the early 1980's. It initially contained 396 tasks, expanded to 610 tasks, and took 320 work-days to complete. The original average staff productivity was approximately 0.27 tasks per man-day.

Project #2 was also a real software project developed in the 1980's. It contained 1866 tasks and took 362 work days to complete. The original average staff productivity was approximately 0.37 tasks per man-day.

High and low anchors were selected for each project and assigned a color code (BLUE BLACK, PINK, PURPLE) for identification as depicted in Table 2.1. The anchors were based on

factors of the original overall average staff productivity achieved in each project. The multiples selected were based on Boehm's work regarding software cost estimation accuracy as a function of the software life-cycle phase. His work suggests that by the detailed design and specification phase, software estimation should be accurate within a factor of 1.25 in either direction (Boehm, 1984).

Table 2.1 PROJECT COLOR CODES AND ANCHOR ASSIGNMENTS

Project	Color	Anchor	Original	Factor	Anchor
Project 1	BLUE	LOW	0.27	0.80	0.18
Project 1	BLACK	HIGH	0.27	1.25	0.41
Project 2	PINK	LOW	0.37	0.80	0.25
Project 2	PURPLE	HIGH	0.37	1.25	0.55

C. EXPERIMENTAL SUBJECTS

The subjects for this experiment consisted of students from two segments of an IS-4300 Software Engineering and Management course at the Naval Postgraduate School. Segment one consisted of 17 students and segment two consisted of 17 students, for a total population size of 34. Table 2-3 lists relevant demographics concerning the subjects. There were no significant deviations between the groups and none of the subjects had any significant experience in software project management.

Table 2.2 SUBJECT DEMOGRAPHICS BY GROUP ASSIGNMENT

Characteristic	All	Group 1	Group 2	Group 3	Group 4
Number of					
Subjects	34	8	9	8	9
Males	28	7	6	8	7
Females	6	1	3	0	2
Average					
Age	34	31	35	34	34
Undergrad.	11	4	14	12	10
Work Exp.	10	8	10	9	12
Fam. Comp.	6	7	6	7	6
Hrs. Comp.	11	14	7	13	11

Key: Age = Age of subjects (years)
 Undergrad. = Years since completing undergraduate ed.
 Work Exp. = Full time work experience (years)
 Fam. Comp. = Familiarity with computers (1=low, 9=high)
 Hrs. Comp. = Hours per week spent using computers

In order to randomize the sample population and assign each subject to one of the experimental groups listed in Table 2-1, the following matched sample procedure was used.

An alphabetical list for each segment was used along with a standard table of random digits to perform the randomization. Appendix A includes the sample population randomizing worksheet used for the experiment. Column A is 5 digit random numbers, taken from a standard table of random numbers, assigned to the alphabetical listing of the students in both sections. Column B is a listing of the students in ascending numerical order according to their assigned random numbers. The four experimental group assignments, Group 1 BLACK/PINK, Group 2 PINK/BLACK, Group 3 PURPLE/BLUE, Group 4 BLUE/PURPLE,

were then repeatedly listed in column C, assigning one of the project combinations to each student. To ensure each of the four project combinations were represented in the protocol analysis, the same randomizing procedure was applied to the last four students on the worksheet selected to use tape recorders.

Although the subjects were not practicing software project managers, the amount of training completed in the curriculum and experience with similar software management experiments leads to the assumption that the results of the experiment and the conclusions would be representative of the cognitive aspects regarding decision making in such tasks. This is supported by Remus's (1986) experiments finding no significant differences between graduate students and similarly educated business managers in making production scheduling decisions. Although software project management decisions are somewhat different from production scheduling decisions, they are similar enough to apply his findings to the assumption that graduate students are acceptable surrogates in this thesis's experimental investigation.

To set the appropriate motivating environment, students were informed that the experiment was an integral part of the Software Engineering Management course they were concurrently taking. Class time was formally allocated for the experiment and ten percent of their final grade was dependent on their attendance and quality participation.

D. EXPERIMENTAL SETTING

1. Software

The SDM gaming interface includes the Dynamo simulation files as well as the Dynex Executive Interface files which allow the model designer to interface with the Dynamo simulation language. The objective was to assimilate a set of files which capture data unobtrusively while allowing the experimental subject to simply start and play the gaming interface without having to learn the simulation language. A quick overview of the major files used in the SDM gaming interface follows.

Three files controlled the simulation's input/output interface (BATCH.BAT, PROJ.DNX, MENU.EXE) and three files produced the necessary output reports (REPORT1.OUT, REPORT2.OUT, REPORT3.OUT). Appendix B contains a listing of all these files.

BATCH.BAT can be thought of as the traffic monitor for the simulation interface. It started the appropriate Dynamo files controlled by PROJ.DNX, had the three reports generated, and called MENU.EXE to supervise the display of the reports every time interval.

PROJ.DNX was the Dynex control file used to direct the subject's input of the staff's average productivity after every time interval. Before the first interval began, some important points to remember concerning the simulation and the

project's initial estimates report were displayed. This is the first report shown to the subject and it contained the anchor, *Project Productivity*. Thereafter PROJ.DNX was only used to accept the subject's productivity estimate input into the model. All subsequent reports were displayed by MENU.EXE.

MENU.EXE provided a menu interface for the subject to selectively display the three available reports, *Initial Estimates Report*, *Project Performance Report*, and *Project Status Report*. The subjects were given the option of examining any or all of the three reports and could return to previously viewed reports within the same time period as desired. This file also contained a routine to capture all the data generated by the subject and the model after every time interval.

2. Reports Provided at each Time Interval

REPORT1.OUT contains the format for the *Initial Estimates Report*. Table 2.3 shows the information displayed in the *Initial Estimates Report*. This report displayed the initial estimates for the project as forecast by management at

Table 2.3 INITIAL ESTIMATES REPORT

1) Project Size	(Tasks)
2) Schedule Duration	(Work-days)
3) Project Productivity	(Tasks/person-days)

the beginning of the simulation and contained the anchor, *Project Productivity*, and the subject's goal, *Schedule Duration*. This report was based on historical data, and was not updated over the project's lifecycle.

REPORT2.OUT contains the format for the *Project Performance Report*. This report was generated by the project staff every 40 work-day interval and was based on their own work records. Table 2.4 shows the information displayed in the *Project Performance Report*.

Table 2.4 PROJECT PERFORMANCE REPORT

1) Elapsed Time	(Work-days)
2) # of Tasks Completed to Date	(Tasks)
3) % Development Completed to Date	(Percents)
4) % Testing Completed	(Percents)
5) Person-days Expended to Date	(Person-days)
6) Current Staff Size	(Fulltime staff)
7) Reported Productivity	(Tasks/person-days)

REPORT3.OUT contains the format for the *Project Status Report*. This report was generated by the project staff every 40 work-day interval and was a forecast based on their last *Project Performance Report*. Table 2.5 shows the information displayed in the *Project Status Report*.

Table 2.5 PROJECT STATUS REPORT

1) Elapsed Time	(Work-days)
2) Estimated Total Project Size	(Tasks)
3) Estimated Total Person-days	(Person-days)
4) Estimated Total Project Duration	(Work-days)

3. Information Provided to Subjects

Two days prior to the experiment, the subjects were introduced to the exercise with a lecture describing the important concepts related to the simulation and gaming interface. The 60 minute presentation included a general description of the exercise, terms used and definitions, the subjects' role in the simulation, their objective, and their ability to influence the project in order to achieve their objective. Since there was no straight forward calculation that would yield the correct answer until the final project statistics were known, the training session gave insight into some of the considerations that should go into the subjects' revised productivity estimations and a reminder that early reported project statistics generally follow the budgeted and not the actual progress of the project. The subjects were also reminded to independently perform the exercise to the best of their ability in order to receive full credit towards their Software Engineering Management course.

On the day of the experiment, each subject was given an exercise package containing a written instruction set, two project documentation sheets, three questionnaires, and one 5.25 inch floppy diskette containing the appropriate project simulation files for that individual's group.

The written instruction set contained information about software project management, the simulation gaming interface, and microcomputer instructions needed to perform

the experiment. Included in the instruction set was a description of the environment, purpose, scope, goals, considerations, rules and procedures to be used for the exercise. A documentation sheet was provided for each simulation so the subject could write down his or her productivity estimates at each time interval for referral and verification. Appendix C contains a copy of the instruction set and a sample documentation sheet.

Each subject was also given a questionnaire to be completed after each simulation and a questionnaire to be completed after the entire experiment. The purpose of the questionnaires was to document the subjects' perceptions of the exercise and gain the necessary sample population characteristics needed for statistical analysis. Appendix D contains a copy of the two questionnaires.

The subjects were given 20 minutes to read the instruction set and understand the experiment procedure. Any questions the subjects had were then answered before proceeding to the microcomputer labs.

The experiment was conducted on 16 microcomputers in two separate labs. Each lab was supervised by a lab attendant familiar with the exercise software, procedures, and lab equipment. The subjects performed each project simulation in accordance with the instruction set, in the order specified for their project group.

After starting the appropriate project simulation, the subject followed the online instructions, referring to the written instruction set as needed. For each time interval, the three reports were provided, a decision regarding the staff's productivity was made, and that estimate was recorded on the documentation sheet and entered into the SDM. When project duration time ceased to increase, indicating the Implementation and Testing phases were complete, the subject had completed the project.

After a subject completed a project, the lab attendant verified the project complete, checked the documentation sheet, and insured the appropriate questionnaire was filled out. The subject then continued the exercise with the second project until it was verified complete by the lab attendant and the appropriate questionnaire filled out. After both project simulations were completed the subject filled out the overall exercise questionnaire and handed in the entire exercise package to the lab attendant.

The four subjects selected to take part in the protocol analysis performed the exercise in a microcomputer lab isolated from the other subjects. Each of the four was given a cassette tape recorder and told to record their thoughts after each time interval for both project simulations. They were asked to give particular attention to the methodology they used to calculate their revised productivity estimates. Otherwise, these subjects received the same

training and performed the experiment just as their counterparts did.

E. DEPENDENT MEASURES

Two dependent measures were used to test the hypotheses:

1) Deviations between productivity estimates made by subjects given a low initial estimate of average productivity and productivity estimates made by subjects given a high initial estimate of average productivity for the same project were used to test H_1 .

2) Deviations between the performance of subjects given a low initial estimate of average productivity and subjects given a high initial estimate of average productivity for the same project were used to test H_2 . Performance was measured by the number of work-days it required a subject to successfully manage a project from start to end.

III. EXPERIMENT RESULTS AND ANALYSIS

A. MAIN EXPERIMENT

The data collected from the experiment contains the subjects' estimates of average staff productivity for each time interval and the project durations, for each project, from 34 subjects.

The productivity estimates were analyzed through a multivariate analysis of variance (MANOVA) model with repeated measures suitable for within subjects designs (Winer 1971). The analyses were performed using the General Linear Models procedure in SAS (SAS, 1987).

Several of the subjects completed their projects prior to the sixth time interval (240 Work-days). To prevent missing variables from skewing the results of the analysis, only productivity estimates made for the first five time intervals (40 to 200 Work-days) were used in the analysis of the subjects' productivity estimates.¹ Time interval 0 is not included because the subjects were not given the option to change the initial estimate of staff productivity until after the first 40 work-day time interval.

¹ Analyses of the data over the first six and seven time intervals provided similar results and conclusions.

1. Subjects' Productivity Estimates

The mean's of the average staff productivity estimates made by the subjects for the first five time intervals are grouped by the anchor given and plotted in Figure 3-1 for Project 1 and Figure 3-2 for Project 2.

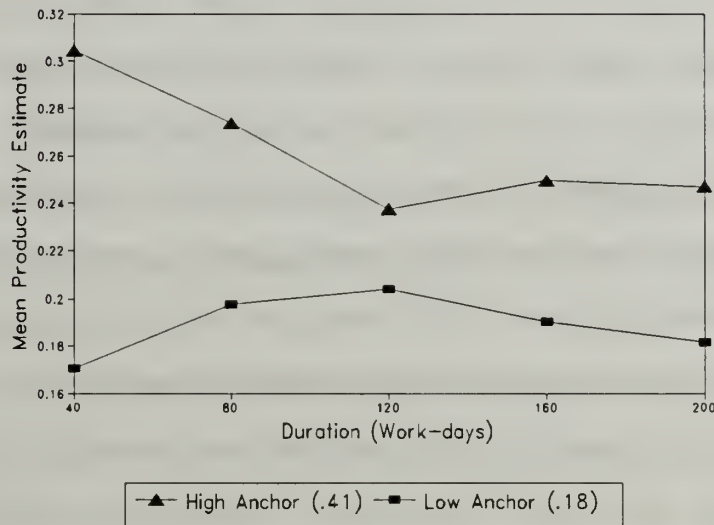


Figure 3-1 Project 1 Mean Productivity Estimates

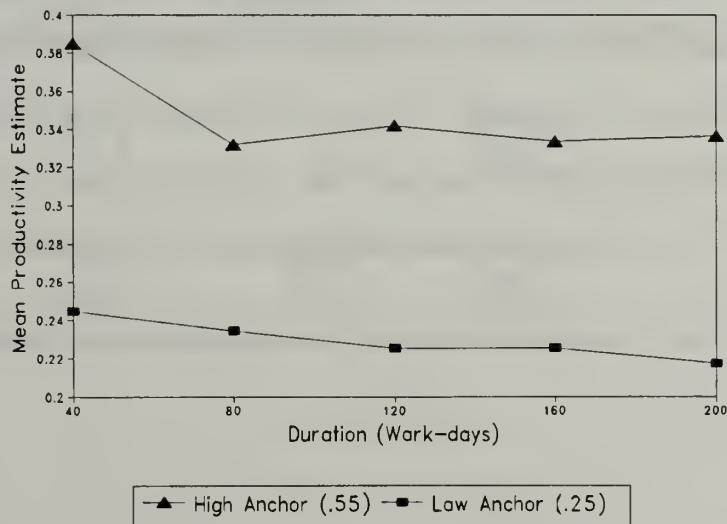


Figure 3-2 Project 2 Mean Productivity Estimates

A visual inspection of the plots suggests the two groups' productivity estimates for both projects appear to be parallel. Since the only difference between the two groups was the anchor given, the lack of convergence suggests the subjects were somehow influenced by the initial estimate of the staff's productivity when making their own productivity estimations.

Another observation is that the subjects were inclined to be pessimistic about the initial estimates provided, regardless of the which anchor was given. Subjects revised their estimates down and then stabilized somewhere below the original average productivity for the project.

Table 3-1 summarizes the MANOVA results for "between-subjects effects" and "within-subject effects".

Table 3-1 RESULTS OF REPEATED MEASURES TESTS

Source of Variation	S.S	Degrees of Freedom	F-Value	p
Between-Subjects				
Anchor	0.7524	1	7.76	0.0070
Project	0.3207	1	3.31	0.0737
Subjects within cells	6.2084	64		
Within-Subjects				
Time	0.0373	4,61	2.23	0.2575
Time*Anchor	0.0099	4,61	2.23	0.1439
Time*Project	0.0018	4,61	0.45	0.2596

a. Between-Subjects Effects

(1) *Different Anchors Effect.* The null hypothesis states that the productivity estimations provided by subjects given different anchors are not significantly different over time. Referring to Figures 3-1 and 3-2, this is the same as saying that the two lines depicting the mean productivity estimates for each anchor group are identical. The test yielded a p-value of 0.007, thereby rejecting the null hypothesis. The rejection of the null hypothesis demonstrates that the productivity estimates made by subjects in different anchor conditions are indeed significantly different. Thus, H_1 is supported.

(2) *Different Projects Effect.* The null hypothesis states that the productivity estimations provided by subjects given different projects are not significantly different over time. Referring to Figures 3-1 and 3-2, this is the same as saying that the two lines depicting the mean productivity estimates for each project are the same. The test yielded a p-value of 0.073, thereby rejecting the null hypothesis. The rejection of the null hypothesis demonstrates that the productivity estimates made by subjects differed from one project to another.

b. Within Subjects Effects

(1) *Time Effect.* The null hypothesis states that the productivity estimations provided by subjects did not vary significantly over time. Referring to Figures 3-1 and 3-2, this is the same as saying that the lines depicting the mean productivity estimates for each anchor group are horizontal. The test yielded a p-value of 0.2575, preventing the rejection of the null hypothesis. Therefore the lines cannot be described as being significantly non-horizontal. Thus, subjects' productivity estimates did not change significantly over time.

(2) *Time and Different Anchors Effect.* The null hypothesis states that the productivity estimations provided by subjects given different anchors did not vary significantly over time. Referring to Figures 3-1 and 3-2, this is the same as saying that the two lines depicting the mean productivity estimates for each anchor group are parallel. The test yielded a p-value of 0.1439, preventing the rejection of the null hypothesis. Therefore the lines cannot be described as being significantly non-parallel. Thus, the productivity estimates made by subjects in different anchor groups did not change significantly over time.

(3) *Time and Different Projects Effect.* The null hypothesis states that the productivity estimations provided by subjects given different projects did not vary

significantly over time. Referring to Figures 3-1 and 3-2, this is the same as saying that the lines depicting the mean productivity estimates for each project are parallel. The test yielded a p-value of 0.2596, preventing the rejection of the null hypothesis. Therefore the lines cannot be described as being significantly non-parallel. Thus the productivity estimates made by subjects in different projects did not change significantly over time.

2. Subjects' Performance

Tables 3-2 and 3-3 list the subjects' performance data as determined by the mean project completion times (in Work-days). The tables are organized by anchor and the order in which the project was simulated.

A quick inspection of the mean completion times for a given anchor reveals the order in which the projects were performed did not significantly effect the subjects' performance. However, it is interesting to note that the mean completion times were lower, albeit not significantly, whenever project 2 was performed first, regardless of which anchor was used.

The subjects' performance results also suggest that Project 1 was more difficult to manage than Project 2. Project 1's mean duration times were all above the duration goal of 320 Work-days, where as Project 2's means were mostly below its goal of 362. The performance difference between

projects could be attributed to Project 1's increasing number of tasks required, from the initial estimate of 396, to 610 tasks by the end of the project, where as Project 2's tasks required remained steady throughout the project.

Table 3-2 PROJECT 1 SUBJECT PERFORMANCE DATA

Anchor	Project 1 Order	N	Completion Times in Work-days		
			Goal	Mean	Std Dev
High(.41)	1st	9	320	368.9	89.1
High(.41)	2nd	8	320	346.9	78.3
Low(.18)	1st	9	320	515.0	56.4
Low(.18)	2nd	8	320	487.5	64.1

Table 3-3 PROJECT 2 SUBJECT PERFORMANCE DATA

Anchor	Project 2 Order	N	Completion Times in Work-days		
			Goal	Mean	Std Dev
High(.55)	1st	9	362	365.0	9.3
High(.55)	2nd	8	362	338.1	40.6
Low(.25)	1st	8	362	331.3	53.6
Low(.25)	2nd	9	362	345.6	34.4

Table 3-4 lists the results of a General Linear Models Procedure testing for effects on subject performance by project.

Table 3-4 SUBJECT PERFORMANCE TESTS BY PROJECT

Source of Variation	S.S	Degrees of Freedom	F-Value	p
Project 1				
Anchor	17515.9	1	32.71	0.0001
Order	5191.7	1	0.97	0.3326
Anchor*Order	63.7	1	0.01	0.9138
Project 2				
Anchor	1337.6	1	0.93	0.3431
Order	267.2	1	0.19	0.6698
Anchor*Order	3488.6	1	2.42	0.1304

a. Different Anchors Effect

The null hypothesis states that different anchors had no effect on subject performance (as measured by project completion times in work-days). In other words, was subject performance affected by the anchor given. For Project 1 the test yielded a p-value of 0.001, thereby rejecting the null hypothesis. For Project 2 the test yielded a p-value of 0.3431, preventing the rejection of the null hypothesis. Therefore for Project 1, subject performance was significantly affected by the anchor, while for Project 2, subject performance was not significantly affected by the anchor. Thus, H_2 cannot be fully supported.

b. Different Order Effect

The null hypothesis states that the order in which a project was performed had no effect on subject performance. The tests yielded p-values of 0.3326 and 0.6698, preventing a rejection of the null hypothesis for both projects. Therefore, subject performance was not significantly affected by the order in which a project was given.

c. Anchor in Different Order Effect

The null hypothesis states that for a given anchor the order in which a project was performed had no effect on subject performance. The tests yielded a p-value of 0.9138 and 0.1304, preventing a rejection of the null hypothesis for both projects. Therefore, subject performance was not significantly affected by the order in which a project with the same anchor was performed.

B. COGNITIVE MODELS

Four subjects were given tape recorders to record their thoughts for a simple protocol analysis. One of the subjects failed to operate the tape recorder correctly and the transcript was never recorded. Therefore only three transcripts were used to perform the protocol analysis.

Project 2's transcripts were used to make the analysis because the complexity level of Project 1 caused subjects great confusion and no discernable protocol analysis could be made from the transcripts provided. Of the three transcripts

from Project 2, two were with high anchors(.55) and one was with the low anchor(.25). Subjects recorded their thoughts after each time interval. A sample time interval from Subject 1's transcript contained the following:

Day 200. I'm still satisfied with the reports I'm getting from the team. Project size and total duration are still on track as far as the numbers I have available to me. The team is still reporting a slow increase in productivity. I still believe we are a little short in required staff, so I bumped my figure down, but only by a tenth of a percent. My staffing has been stable now at around 20.5 personnel for a considerable time and I want to keep it there because I think we can finish the project with this size team.

The simple protocol analysis consisted of breaking down the transcripts into "semantic elements" as described by Bouwman (1983). The elements are classified as either an "item" of information, an "operator" on an item, or the "result" of an operator on an item. These elements (item, operator, result) are then formed into functional groups by linking an operator element to the item it uses to produce a result. Functional groups were found by examining the transcripts for repetitive operations used to gain information and make conclusions.

All three subjects used comparisons to gain information and establish trends over time. Each functional group was composed of two items which the operator "compared" to form a result. Subjects would compare an item's current value with its original or last reported value, or a perceived needed value, to establish trends. The trends were then used to

formulate a direction to move their revised estimate of productivity.

Tables 3-4, 3-5, and 3-6 display each subject's protocol analysis, composed of the functional groups each subject had in common, and their resultant trends. A minus sign (-) indicates that Item1 is less than Item2, a plus sign (+) indicates that Item1 is greater than Item2, and an equals sign (=) indicates the Items were equal. If there is no indicator present, the subject did not report making that observation for the time period.

The revised productivity estimates made by the three subjects are plotted in Figure 3-3 for comparison with their individual protocol analyses in Tables 3-4, 3-5, and 3-6.

All three subjects' mental models revolved around their perception of the staff size needed to complete the project within the scheduled duration time. The subjects continually tried to manipulate the project staff size with their estimates of productivity in order to achieve or maintain a desired staff level.

The subjects were keenly aware of the time lags involving productivity and staff level changes. When an influx of new staff personnel was achieved, they recognized the resulting drop in productivity as training and familiarization overhead

Table 3-4 SUBJECT 1, PROTOCOL ANALYSIS OF HIGH ANCHOR

Item1	Item2	Time Interval (Work-days)							
		40	80	120	160	200	240	280	320
Proj. size	original	=	=	=	=	=	=	=	=
Est. duration	original	=	=	=	=	=	=	=	+
Team prod.	last reported	-	-	+	+	+	+	+	+
Team prod.	original	-	-	-	-	-	-	-	-
Team prod.	last est.	-	-	-	-	-	-	-	-
Team size	last	+	+	+	=	+	+	=	=
Team size	needed	-	=	=	-	-	-	-	-
REVISED PROD.	last	-	=	+	-	-	-	-	-

Table 3-5 SUBJECT 2, PROTOCOL ANALYSIS OF HIGH ANCHOR

Item1	Item2	Time Interval (Work-days)							
		40	80	120	160	200	240	280	320
Proj. size	original								
Est. duration	original						+	+	+
Team prod.	last reported	-	-	+	=	-	+	+	+
Team prod.	original	-							
Team prod.	last est.								
Team size	last							+	=
Team size	needed		=	=	-	=	-	=	=
REVISED PROD.	last	-	=	-	-	=	-	+	=

Table 3-6 SUBJECT 3, PROTOCOL ANALYSIS OF LOW ANCHOR

Item1	Item2	Time Interval (Work-days)							
		40	80	120	160	200	240	280	320
Proj. size	original		=		=	=		=	
Est. duration	original	-	-	-	-	=	=		
Team prod.	last reported				+	+		+	
Team prod.	original	-	-						
Team prod.	last est.		-	-	=	+		+	
Team size	last								
Team size	needed		-	=	=	=	=		
REVISED PROD.	last	-	-	=	+	+	+	+	

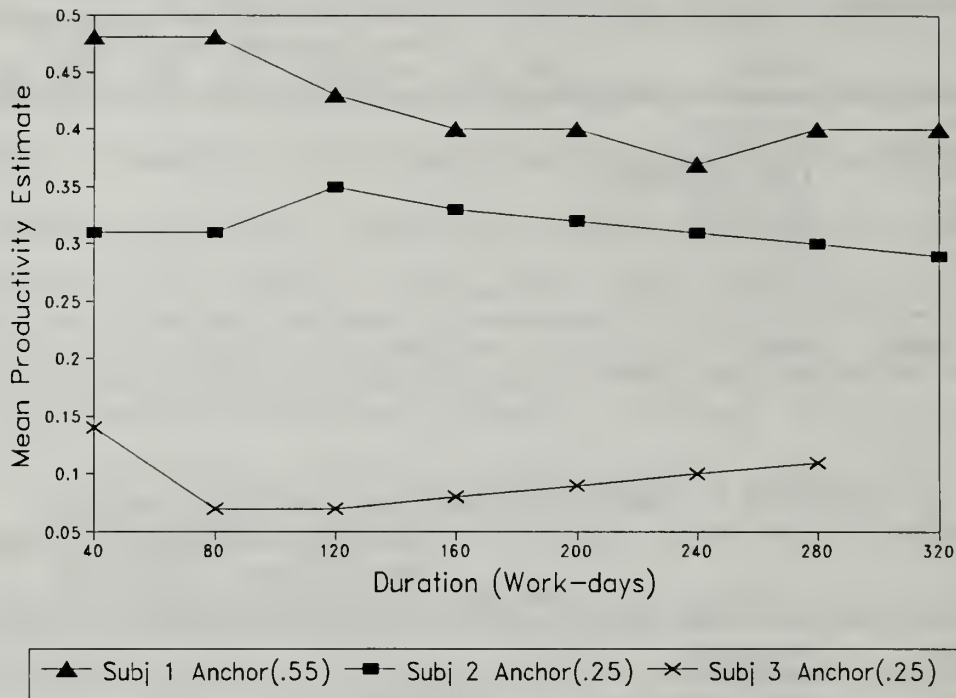


Figure 3-3 Revised Productivity Estimates from Subjects used in Protocol Analysis

and waited for the productivity to stabilize before making further adjustments. For example, Subject 1, time interval 80:

...team size has doubled in the last 40 days and I'm afraid that if I continue to revise down the productivity measures I'm going to get exponential staff growth and the negative payoff for training the new people is going to kill me. I want to attempt to keep the staff size constant by holding my productivity estimate constant and give the team a chance to climb up the learning curve.

And when they perceived the project to be ahead of schedule, they raised the productivity estimate to reduce the staff size as needed. For example, Subject 3, time interval 200:

...all those people we hired are starting to produce now and their familiarity with the project is increasing. I'm going to increase my old estimate from 160, from .08 to .09 and see if it works.

As was previously observed in the main experiment, the three subjects initially revised their productivity estimates down from the anchor, regardless of whether the anchor was high or low. The protocol analysis shows this as a desire to front load the staff with manpower in an attempt to "get ahead of the game." For example, Subject 3, time interval 80:

...I'm going to drop down the productivity from the reported of .11, down to .07, to try and front load the project with people right from the start and get them trained up and get them rolling so that we're not adding people piecemeal throughout the project. Hopefully that will jump start this thing.

By trying to increase the staff size early, they were hoping to weather the lost productivity caused by training and familiarization in order to reap the benefits of a larger staff over the remainder of the project lifecycle.

IV. CONCLUSIONS

A. SUMMARY OF OBJECTIVES

The objective of this thesis was to investigate how managers make decisions in complex dynamic environments where dynamic decision making is involved, and what effect this process had on their performance. Chapter I (section B.2) discussed how decision makers turn to simple cognitive heuristics, such as anchoring-and-adjustment, to simplify complex decision making. Earlier studies had found the use of such simple judgmental operations can result in cognitive biases leading to dysfunctional performance, but Hogarth had argued this was a result of the discrete and static nature of the experiments and further study was needed in dynamic experimental settings.

Chapter I (sections B.1 and B.3) explained how software project management was not only in a dynamic environment, but was also a dynamic decision making process where past estimations affect the present environment in which current estimates must be made. Therefore, given that software project management is in a complex dynamic environment involving dynamic decision making, do managers use cognitive heuristics such as anchoring and adjustment to simplify the decision

making processes? And if so, does it have an effect on performance?

B. SUMMARY OF RESULTS

Two hypotheses were stated in Chapter I (section C) and tested in accordance with Chapter II. Chapter III discussed the analysis of the data and found the following results.

1. H_1 : Anchoring-and-Adjustment is Used

Although the simple protocol analysis did not detect the subjects consciously anchoring their estimates to the initial estimate provided, the statistical evidence does suggest the anchoring-and-adjustment heuristic was used by the subjects in their decision making. The analysis of variance test showed a significant difference in subjects' productivity estimations depending on the anchor provided ($F=7.76$, $p=0.0070$). Thus, H_1 is supported.

It was also shown that the subjects did not vary their estimates significantly over time ($F=2.23$, $p=0.2575$), suggesting they did not abandon the anchor over the project's lifecycle.

2. H_2 : Performance is Affected

For Project 1, the analysis of variance test showed a significant difference in the subjects' performance depending on the anchor provided ($F=32.71$, $p=0.0001$). The mean completion times suggest that a high initial productivity estimation

will result in better performance than when given a low initial estimate.

However for Project 2, the statistical data does not support the hypothesis ($F=0.93$, $p=0.3431$). Therefore the statistical analyses of variance in performance due to the anchor given was inconclusive and H_2 cannot be fully supported.

One possible explanation for the mixed performance results between the two projects is that more anchoring-and-adjustment bias was introduced into the estimations made by subjects in the more complex project (Project 1), than in the less complex project (Project 2). It was evident from the transcripts provided for the protocol analysis that the students had a much clearer mental model of Project 2, than of Project 1. This may have caused the subjects' to abandon their ambiguous mental model of Project 1's dynamic environment and rely more on the project heuristics to make their estimates, resulting in dysfunctional performance.

C. IMPLICATIONS OF RESULTS

The results of the experiment provide several implications for managers making dynamic decisions in complex dynamic environments, specifically those in support of software development projects. The anchoring-and-adjustment heuristic was used by project managers to simplify decision making in software project management with dynamic decision making,

expanding upon the findings of Ronan (1990) in this area of research.

Although the analysis of the subjects' performance provided inconclusive results, there was some evidence of dysfunctional performance when making dynamic decisions using the anchoring-and-adjustment heuristic. Managers did not make significant adjustments from the anchor as the project lifecycle progressed. The use of a dynamic environment as Hogarth had suggested, was not enough to enable the subjects to use anchoring-and-adjustment affectively when dynamic decision making was added to the process.

The results show that managers do not consciously anchor their revised estimates on initial estimates. Project managers must be made aware of the effect anchoring-and-adjustment and the initial estimates have on the development process. Either the process of making revised estimates must be improved allowing managers to form better mental models of the dynamic system so simple cognitive heuristics are no longer needed or the initial estimates must be made with the anchoring-and-adjustment phenomenon in mind.

D. LIMITATIONS AND FUTURE RESEARCH

Although the experiment simulated real life software projects in a proven simulation model, it is difficult to claim external validity for laboratory-type studies. Remus (1978) indicated that decision making in games and managerial

decision making are similar enough to relate experimental findings to the real world. However, software project management is not a game that is played in one sitting, so comparisons should be limited to the cognitive aspects involved in both settings.

As discussed in Chapter II (section C), a second limitation was the fact that the subjects were not practicing software project managers. Although using graduate students as surrogates in research studies is useful, analyzing the behavior of experienced project managers could lead to more practical and pointed results.

The simple protocol analysis had several limitations. First, three transcripts did not produce enough information to perform a full analysis. There was not enough similarity in the subjects approaches to form an overall protocol governing the decision making processes involved. Secondly, the subjects recorded their thoughts in a discrete nature, packaging the information. Instead of a continuous stream of thoughts representing the decision making processes involved, subjects' "summarized" their decisions making process after each time interval. This severely limited the amount of information captured and biased it towards what the subject believed he or she thought, not what was actually thought, thus, losing the less conscious or believed irrelevant thoughts through refinement. A full protocol analysis with a

greater number of subjects recording their thoughts continuously would provide more significant and noteworthy results.

There are many variations of the experiment which could be tested. One possible change to the experimental design would have each subject simulating each anchor condition on the same project, instead of two separate projects. Special care would have to be taken to prevent biases from being formed because of the order in which an anchor condition was operationalized, but using one project could remove variances in the analysis caused by the differences in experience and ability levels between subjects.

APPENDIX A

SAMPLE POPULATION RANDOMIZING WORKSHEET

Column A			Column B		Column C		
RANDOM#	NAME		RANDOM#	NAME	GROUP #	1ST RUN	2ND RUN
1	15544	ABBOTT	1011	BAKER	GROUP 1	BLUE	PURPLE
2	1011	BAKER	7851	HARMS	GROUP 2	BLACK	PINK
3	47435	BLAKE	8768	PREVOST	GROUP 3	PINK	BLUE
4	91312	BOURQUE	9300	CHUN	GROUP 4	PURPLE	BLACK
5	12775	BOYERS	9402	HURAL	GROUP 1	BLUE	PURPLE
6	31466	BUSCH	11092	DONOHUE	GROUP 2	BLACK	PINK
7	9300	CHUN	11264	ELLIOTT	GROUP 3	PINK	BLUE
8	73582	DAVIS	12775	BOYERS	GROUP 4	PURPLE	BLACK
9	11092	DONOHUE	13810	THUR	GROUP 1	BLUE	PURPLE
10	93322	DOWLER	15544	ABBOTT	GROUP 2	BLACK	PINK
11	80134	DUVALL	21285	KOTHEIMER	GROUP 3	PINK	BLUE
12	11264	ELLIOTT	25594	HAYES	GROUP 4	PURPLE	BLACK
13	2612	EMERY	31466	BUSCH	GROUP 1	BLUE	PURPLE
14	96256	GIBBONS	31797	ZELLMANN	GROUP 2	BLACK	PINK
15	7851	HARMS	43761	RAGAN	GROUP 3	PINK	BLUE
16	25594	HAYES	43847	STENZOSKI	GROUP 4	PURPLE	BLACK
17	65358	HOWE	47435	BLAKE	GROUP 1	BLUE	PURPLE
18	9402	HURAL	53308	PARRISH	GROUP 2	BLACK	PINK
19	97424	JENNINGS	65358	HOWE	GROUP 3	PINK	BLUE
20	80712	KOHLHEIM	66433	VAUGHN	GROUP 4	PURPLE	BLACK
21	21285	KOTHEIMER	73582	DAVIS	GROUP 1	BLUE	PURPLE
22	53308	PARRISH	75137	PAYLOR	GROUP 2	BLACK	PINK
23	75137	PAYLOR	80134	DUVALL	GROUP 3	PINK	BLUE
24	8768	PREVOST	80712	KOHLHEIM	GROUP 4	PURPLE	BLACK
25	43761	RAGAN	81392	VANHOOK	GROUP 1	BLUE	PURPLE
26	43847	STENZOSKI	91312	BOURQUE	GROUP 2	BLACK	PINK
27	13810	THUR	92612	EMERY	GROUP 3	PINK	BLUE
28	81392	VANHOOK	93322	DOWLER	GROUP 4	PURPLE	BLACK
29	66433	VAUGHN	96256	GIBBONS	GROUP 1	BLUE	PURPLE
30	31797	ZELLMAN	97424	JENNINGS	GROUP 2	BLACK	PINK

Four Students Used in Protocol Analysis

31	27082	DICKISON	27082	DICKISON	GROUP 1	BLUE	PURPLE
32	45586	ESTRADA	45586	ESTRADA	GROUP 2	BLACK	PINK
33	70653	LINDSEY	47452	RICHADSON	GROUP 3	PINK	BLUE
34	47452	RICHADSON	70653	LINDSEY	GROUP 4	PURPLE	BLACK

APPENDIX B

BATCH. BAT

```

echo off
CLS
init 1H
:GRAPHICS
bat /N /p /s
smlt PROJ1 -go = -prs = -ls -ns -plm 6 -bw
-top dynex PROJ1 -in PROJ1.STT -sc -ls -plm 6 -bw
    smlt PROJ1 -gm = -ns -plm 6 -bw
    rep PROJ1 INTRVAL -outf INTERVAL.OUT -t -bw >NUL
    rep PROJ1 REPORT1 -outf REPORT1.OUT -t -bw >NUL
    rep PROJ1 REPORT2 -outf REPORT2.OUT -t -bw >NUL
    rep PROJ1 REPORT3 -outf REPORT3.OUT -t -bw >NUL
    rep PROJ1 -bw >NUL
    infoofb 1
anchor 1H
-2~1 **** PROCEED WITH NEXT SIMULATION ****
    BAT CLS
    BAT COLOR \1F
    BAT BEGTYPE
*****
*
*
*   1.  Determine your estimate for the project team's
*
*   average productivity (in task/person-days) and
*
*   WRITE IT ON THE DOCUMENTATION SHEET.
*
*
*
*   2.  TO CONTINUE PRESS <ENTER>.
*
*
*****
END
bat /p /s goto -top
-%0~1
-$%0$1
-%0%11 beep goto -top1
-on.error-
    if %R > 82 if %R < 90 type !! Floating Point Error !! |goto
-Calcul.
    Cls beep type Unexpected batch file error %R in line %L
exit

```

PROJ.DNX

```
if #tm<0.9 then
d totmdl=0.18
display clear
```

Important Points to Remember !!!!!!!!!!!

- You are not allowed to discuss this exercise with anyone other than a lab attendant. Please refrain from discussing this with other class members until they have completed the project.
- The system will run through the first simulation period (40 work-days) and provide you with 3 reports. At the end of each reporting period, you will have an opportunity to revise the estimated productivity (in tasks/person-days).
- The system is slow due to reading and writing from the floppy disk. There will be approximately 1.5 minutes of disk grumblings inbetween reporting periods so PLEASE BE PATIENT! and wait for the simulation prompts.
- Make your changes to the productivity estimate on the documentation sheet provided and then on the screen.
- A LAB ATTENDANT MUST VERIFY YOUR FINAL RESULTS!
- GOOD LUCK! Press <ENTER> to continue.

```
dendq
choice 1
cend 1/1
display clear
```

INITIAL ESTIMATES REPORT

ELAPSED TIME =	=	=	=	=	=	=	=	=	=	> 0	Days
Project Size									396		Tasks
Schedule Duration									320		Days
Project Productivity									0.18		Tasks/person-days

The productivity estimate for the first 40 work-days will be based on the initial estimate of 0.18 tasks/person-days.

Press <ENTER> to continue.

```
dendq
choice 1
cend 1/1
d ASSPRD=0.18
else
choice 1
cend 1/1
display clear
```

INPUT YOUR ESTIMATE OF PRODUCTIVITY IN TASKS/PERSON-DAYS

1) Press <ENTER> to maintain your last productivity

***** OR *****

2) Enter your new estimate of productivity (in
tasks/person-days) and press <ENTER>

Your last productivity estimate was =

```
dendq
dq ASSPRD=0<1
display clear
```

!!!!!!! WARNING !!!!!!!

Make sure that you have written down your
estimate on the project documentation sheet
before continuing with the simulation.

This is your final chance to change the estimated
productivity. Press <ENTER> to keep the same
estimate or enter a new estimate and then press
<ENTER>.

The updated estimate of productivity is =

```
dendq
dq ASSPRD=0<1
end
display clear
```

It will take approximately 1.5 minutes to crunch
40 work-days ... please standby.

```
dend
```

MENU.EXE

C SOURCE CODE

```
#include      <stdio.h>
#include      <se.h>
#include      <dos.h>
#include      <ctype.h>

#define       REPORT1  "report1.out"
#define       REPORT2  "report2.out"
#define       REPORT3  "report3.out"
#define       MAXLINE  80

main(argc, argv)
    int argc;
    char *argv[];
{
    char      ch;

    if (argc<2) {
        printf("\n need project no.");
        exit(0);
    }
    off_cursor();
    dump_info(argv[1]);

    for(;;) {
        cls();
        box(0,0,23,79);
        set_cursor(4,22);
        printf("Please enter a number (1-4)");
        set_cursor(10,22);
        printf("1.    View Initial Estimates Report ");
        set_cursor(12,22);
        printf("2.    View Project Performance Report");
        set_cursor(14,22);
        printf("3.    View Project Status Report");
        set_cursor(16,22);
        printf("4.    Provide New Productivity Estimate");

        ch=getch();
        log(ch, argv[1]);
        if(ch=='1')
            readtext(REPORT1);
        else if(ch=='2')
            readtext(REPORT2);
        else if(ch=='3')
            readtext(REPORT3);
        else if(ch=='4')
```

```

        break;
    else {
        set_cursor(24,0);
        printf("Please enter a number between 1-4. Strike any
key to          continue");
        getch();
    }
}
on_cursor();
}

dump_info(proj_no)
char      proj_no[];
{

    char      outfile[FILESIZE];
    float      dat;
    double result;
    FILE      *fi, *fo, *fopen();

    strcpy(outfile, OUTFILE);
    strcat(outfile, proj_no);

    if((fi=fopen(INFILE, "r"))==NULL) {
        printf("\couldn't open %s for read", INFILE);
        exit(0);
    }
    if((fo=fopen(outfile, "a"))==NULL) {
        printf("\couldn't open %s for write", outfile);
        exit(0);
    }
    fprintf(fo, "\n");
    while(!feof(fi)) {
        fscanf(fi, " %f", &dat);
        result=dat;
        fprintf(fo, "%#2.2f ", result);
    }
    fclose(fi);
    fclose(fo);
}

```



```

/*****
* Reads a textfile and prints to screen.
* Input: (char) filename.
* Returns: Nothing.
*****/

readtext(filename)
char filename[];
{
FILE *fi, *fopen();
char line[MAXLINE], *result;
int coodx=3, coody=3;
cls();
box(0,0,23, 79);
if((fi = fopen(filename, "r"))==NULL)
{
    set_cursor(23,0);
    printf("couldn't open %s for r", filename);
}
while(fgets(line, MAXLINE, fi))
{
    if(coodx < 22) /* Still same screen */
    {
        coodx++;
        set_cursor(coodx, coody);
        printf(" %s\n", line);
    }
    else /* Next screen */
    {
        set_cursor(24, 25);
        printf("STRIKE ANY KEY TO CONTINUE");
        getch();
        /*while(!kbhit());*/
        cls();
        box(0,0,23, 79);
        coodx=1;
        coody=1;
        /*printf(" %s\n", line);*/
    }
}
fclose(fi);
set_cursor(24, 25);
printf("STRIKE ANY KEY TO CONTINUE");
getch();
/*while(!kbhit()); */
}
/* Put user trace info for OFB S's in log file */
log(ch, proj_no)
char ch, proj_no[];
{
    struct info userinfo;

```

```

char    logfile[FILESIZE];
int      smc, result;
FILE     *fp, *fopen();
/* Get time */
_dos_gettime(&userinfo.start_time);
strcpy(logfile, "");
strcat(logfile, LOGFILE);
strcat(logfile, proj_no);
if((fp=fopen(logfile, "a"))==NULL) {
    printf("\couldn't open %s for append", logfile);
    exit(0);
}
result=atoi(ch);
if(ch>'0' && ch<'5') {
    fprintf(fp, "\n%c", ch);
    fprintf(fp, "  %#2d:%#2d:%#2d",
        userinfo.start_time.hour,\
        userinfo.start_time.minute,\
        userinfo.start_time.second);
}
fclose(fp);
}

/*****
/      SE.H:  Header file for programs for experiment on *
/                      Anchoring                      *
/*****/

#define LOGFILE          "log"    /*Log file for process
info */
#define INFILE           "interval.out" /*  infile for
data */
#define OUTFILE          "info"    /*infile for data */
#define DATAFILE        "..\\data.dat"
#define RANDFILE         "random.out"
#define FILESIZE         8    /*Size of string for creating
filenames*/
#define CFB_ENDITER      7    /*Signal for end of interval
*/
#define CFB_MINVAL       1    /*Minimum value of expected
keystroke */
#define CFB_MAXVAL       7    /*Max value of expected
keystroke */
#define OFB_ENDITER      2    /*Signal for end of interval
*/
#define OFB_MINVAL       1    /*Minimum value of expected
keystroke */
#define OFB_MAXVAL       2    /*Max value of expected
keystroke */

```

```

/*****
/  Below are defined various structures for use in      *
/  collecting date and time information:  _dos_getdate,  *
/  _dossetdate and _dos_gettime, _dos_settime          *
*****/

#ifndef _DATETIME_T_DEFINED
struct dosdate_t {
    unsigned char day;           /* 1-31 */
    unsigned char month;        /* 1-12 */
    unsigned int year;          /* 1980-2099 */
    unsigned char dayofweek;     /* 0-6, 0=Sunday */
};

struct dostime_t {
    unsigned char hour;         /* 0-23 */
    unsigned char minute;      /* 0-59 */
    unsigned char second;      /* 0-59 */
    unsigned char hsecond;     /* 0-99 */
};

#define _DATETIME_T_DEFINED
#endif

/*****
/  This is the structure for carrying specific information *
/  about the subject.                                     *
*****/

static struct info {
    char name[25];              /* Name of subject */
    int group;                  /* Experimental grp subject belongs
                               to.* 0 = OFB, 1 = CI+TI, 2=CI,
                               3=TI */
    int subject;                /* Subject No. Usually SMC */
    int sequence;               /* Within subjects sequence */

    int phase;                  /* Phase of experiment, i.e.,
                               training, experiment, etc */
    int block;
    int iteration;
    int feed_drule;             /* Type of feedback requested by
                               user. */
    int feed_cons;              /* For writing into logfile */
    int feed_ti;
    int feed_ofb;

    struct dosdate_t date;
    struct dostime_t start_time;
    struct dostime_t end_time;
};

```

REPORT1.OUT

```
REPORT
time=maxtime,
FORMAT="36- "
"INITIAL ESTIMATES REPORT";
Format="10<,46<,58<",PICTURE="Z,ZZ9V"
"ELAPSED TIME  = = = = = =>",tm,"Days";
Format="4<"
"ESTIMATES MADE AT THE START OF THE PROJECT";
FORMAT="4<,44<,58<",PICTURE="ZZZ,ZZZV"
"Project Size",IPRJSZ,"Tasks";
FORMAT="4<,44<,58<",PICTURE="ZZZ,ZZZV"
"Project Duration",TDEV1,"Days";
FORMAT="4<,44<,58<",PICTURE="ZZZ,ZZ9V.99"
"Project Productivity",TOTMD1,"Tasks/person-days";
```

REPORT2.OUT

```
REPORT
time=maxtime,
FORMAT="36- "
"PROJECT PERFORMANCE REPORT";
Format="17<,48<,60<",PICTURE="Z,ZZ9V"
"ELAPSED TIME  = = = = = =>",tm,"Days";
Format="2<,46<,60<",PICTURE="ZZZ,ZZ9V"
"PROJECT STATUS at Time = = = = = =>",tm,"Days";
FORMAT="2<,46<,60<",PICTURE="ZZZ,ZZ9V.99"
"Number of Tasks Reported Complete",
(PDVRC/100)*PJBSZ,"Tasks";
FORMAT="2<,46<,60<",PICTURE="ZZZ,ZZ9V.99"
"% Development (Design & Code) Reported Com-
plete",PDVRC,"Percent";
FORMAT="2<,46<,60<",PICTURE="ZZZ,ZZ9V.99"
"% Testing Reported Complete",PTKTST*100,"Percent";
FORMAT="2<,46<,60<",PICTURE="ZZZ,ZZ9V.99"
"Total Person-days Expended to date",CUMMD,"Person-days";
FORMAT="2<,46<,60<",PICTURE="ZZZ,ZZ9V.9"
"Current Staff Size",FTEQWF,"Fulltime Staff";
FORMAT="2<,46<,60<",PICTURE="ZZZ,ZZ9V.99"
"Average Reported Productivity",
(PDVRC/100)*PJBSZ/CUMMD,"Tasks/person-days";
```

REPORT3.OUT

```
REPORT
time=maxtime,
FORMAT="36- "
"PROJECT STATUS REPORT";;
Format="16<,46<,60<","PICTURE="Z,ZZ9V"
"ELAPSED TIME  = = = = = =>","tm,"Days";;
Format="2<,44<,60<","PICTURE="ZZZ,ZZ9V"
"PROJECT ESTIMATES at Time = = = = = =>","tm,"Days";;
FORMAT="2<,44<,60<","PICTURE="ZZZ,ZZ9V.99"
"Updated Estimate of Total Project Size",PJBSZ,"Tasks";
FORMAT="2<,44<,60<","PICTURE="ZZZ,ZZ9V.99"
"Updated Estimate of Total Man Days",JBSZMD,"Person-days";
FORMAT="2<,44<,60<","PICTURE="ZZZ,ZZ9V"
"Updated Est. of Project Duration (start-end)",
SCHCDT,"Days";
```


APPENDIX C

PRODUCTIVITY ESTIMATION INSTRUCTION SET

Introduction

The exercise you are about to undertake is similar to that of a flight simulator used by a pilot to mimic flying an aircraft from takeoff at point A, to landing at point B. Instead of simulating flight, this computer exercise will simulate the life of a real software project from the start of the implementation phase to the end of testing. In less than an hour you will live through the project's lifecycle. You will play the part of a valuable assistant to the Project Manager. In this simulation your decisions will directly impact on the project's overall cost and completion date.

Project Information

You will be given two separate projects to track, each of them real projects conducted in a real organization. The organization is on the leading edge in its software engineering practices. It uses a customized version of COCOMO which has been calibrated using the organization's extensive database of historical project data. Based on well documented past performance data for software projects of similar size and complexity, a project profile containing the following initial information will be provided for each project:

Project Size (in No. of Tasks)
Schedule Duration (in No. of Work-days)
Project Productivity (in No. of Tasks/Person-days)

A task is a unit of work ... you may think of it as a software module containing 50 lines of code.

Management is adamant about the schedule, so it is imperative the project be completed on time, however, cost is always a priority. Resources are limited and you should

strive to bring the project in on time while keeping costs (in Person-days) to a minimum.

The personnel pool is composed of technically competent and experienced personnel. A database composed of their performance on past projects of similar size and complexity provides the initial project productivity measurement.

Your Objective

Your objective is to come up with the best estimate of the team's expected average productivity (in tasks/person-day). It will be used by the Project Manager to calculate the staff required to complete the project on schedule with the least possible cost.

Specifically, your role will be to track the project's progress using reports produced for you every 40 work-days throughout the project's life. After every 40 work-day period you will make your best estimate of the average productivity required to meet the schedule deadline. Your estimate will be critically important as this information will be used to make the necessary adjustments to the project's staff.

For example, if at some point in the project:

- a. Remaining time = 100 work-days
- b. Remaining tasks = 200 tasks

And you make an estimate of the average productivity:

- c. Estimated productivity = .2 tasks/person-days

Then an estimate of the remaining effort in person-days can be calculated:

- d. $200 \text{ tasks} \div .2 \text{ tasks/person-days} = 1,000 \text{ person-days}$

And remaining effort can be used to calculate the staff required:

- e. $1,000 \text{ person-days} \div 100 \text{ work-days} = 10 \text{ people}$

Since staff size will ultimately determine the project's overall cost and duration, your estimation of the required average productivity is critical to the success of the project.

Your grade for the simulation will be based on your ability to:

First and foremost - bring the project in on schedule.

Secondly - spend as little as possible (in person-days) in the accomplishment of the first objective.

The best way to do this is to provide the most accurate estimation of the Team's actual productivity.

How to Play the Game

** You will be required to provide your estimate of the required actual productivity in tasks/person-days at the beginning of every 40 work-day interval. The simulation will stop to show a menu providing four options:

- 1) View Initial Estimates Report;
- 2) View Project Performance Report;
- 3) View Project Status Report;
- 4) Provide New Productivity Estimate.

** 1) Initial Estimates Report will provide you with the initial estimates of:

Project Size
Schedule Duration
Project Productivity

These estimates are provided by management, based on historical data, and will not be updated over the project's lifecycle. **The Schedule Duration figure is your goal for work-days to completion.**

** 2) Project Performance Report will provide you with information to date on:

Elapsed time
of tasks completed
% development completed
% testing completed
Person-days expended
Current staff size
Average reported productivity

This data is provided by your project team based on their work records and will be updated every 40 work-days. Person-days expended is a running total of project cost. Average reported productivity is the **team's reported** productivity.

- ** 3) Project Status Report will provide you with updated estimates of:

Total project size
Total project cost
Total project duration

This data is provided by your project team based on a projection of their last Project Performance Report to project completion and will be updated every 40 work-days. Total project size may increase due to additional requirements, etc. Total cost and duration are the team's predictions based on information to date.

- ** 4) Provide New Productivity Estimate will allow you to input your estimate and continue with the next 40 work days of the simulation. Your productivity estimate will be used to make staffing decisions for the project over the next 40 work-days. Make sure you write down your estimated productivity for the period on the documentation sheet provided before continuing.

- ** Your task as the assistant to the Project Manager in this simulation can be broken down into 4 steps:

- 1) Look at the 3 reports available to you each period.
- 2) Based on management's estimates and the team's reports, formulate your estimate of the productivity needed to bring the project in on time with the least possible cost.
- 3) Select, Provide New Productivity Estimate, write your productivity estimate for the period on the documentation sheet and then enter it on the computer when prompted. Run the next 40 work-days of the simulation.
- 4) Repeat steps 1 - 3 until the Elapsed Time figure repeats, this signifies you have completed the project.

- ** YOU MUST WORK ALONE. You are not allowed to discuss this exercise with anyone other than a lab attendant. Also, please refrain from discussing this with any member in the other class until they have completed the exercise.
- ** Please follow the guidelines strictly. The system prompts, along with instructions in this booklet, will guide you at every stage.
- ** If you are in doubt about anything, ask for a lab attendant.

Important Considerations

1. The initial project productivity estimate is derived from an extensive database of historical project statistics that this organization has developed and maintained in the last five years. It provides an estimation of the team's average productivity throughout the project's lifecycle.
2. Software is basically an intangible product during the earlier phases of design and coding. It is important to note the reports produced by the project staff may be unreliable initially. That is to say, some inaccuracies will be present in the reports due to estimation difficulties, especially in the early stages of the project. As in any real software project, as time goes on the reports will become more and more accurate, and thus more dependable.
3. The personnel turnover rate is 20% per year.
4. The hiring delay for new employees can take up to 30 work-days. Once new people are hired, the assimilation period for a newly hired employee is typically one month long. This is the time needed to train a new employee in the mechanics of the project and bring him/her up to speed. A new employee (i.e. one that is being trained) is only half as productive as an experienced employee.
5. As the project proceeds, expect the productivity of the team as a whole to increase by around 20-30% due to the learning curve effect.
6. Schedule pressure can cause productivity to go up or down depending on whether the project falls behind or ahead of schedule (e.g., if people perceive that they are falling behind schedule they may be motivated to work longer hours to bring the project back on track).

Exercise Instructions

1. You will simulate two separate software projects today.
2. First, read and understand the entire instruction set before continuing. If you have any questions ask a lab attendant to clarify them.
3. When you are ready to begin insert the floppy disk provided into the A: drive and boot up the computer.
4. At the A> prompt type *PINK* and press <enter> to start the first project simulation.
5. When you have completed PROJECT 1 (when the Elapsed Time figure repeats) have a lab attendant verify your work and then answer the questionnaire for PROJECT 1.
6. After completing the questionnaire, REBOOT YOUR COMPUTER. At the A> prompt type *BLACK* and press <enter> to start the second project simulation.
7. When you have completed PROJECT 1 have a lab attendant verify your work and then answer the questionnaire for PROJECT 2.
8. Answer the questionnaire for the entire exercise and turn in all materials when you are finished.

PROJECT DOCUMENTATION SHEET

YOUR NAME : _____

SMC NO. : _____

Please enter your productivity estimates in the appropriate time period below:

PRODUCTIVITY (tasks/person-day)

Time elapsed - 40 days: _____

Time elapsed - 80 days: _____

Time elapsed - 120 days: _____

Time elapsed - 160 days: _____

Time elapsed - 200 days: _____

Time elapsed - 240 days: _____

Time elapsed - 280 days: _____

Time elapsed - 320 days: _____

Time elapsed - 360 days: _____

Time elapsed - 400 days: _____

Time elapsed - 440 days: _____

Time elapsed - 480 days: _____

***** WHEN YOU ARE DONE, PLEASE CALL FOR A LAB ATTENDANT *****

APPENDIX D

QUESTIONS TO BE ANSWERED AFTER COMPLETING PROJECT 1

1. Describe (in words, numbers, equations, etc) what decision process you followed in deciding the productivity estimate for the project:

2. What helpful hints would you give to someone who was about to begin the simulation you just performed:

3. How helpful was the Initial Estimates Report in your decision making:

1	2	3	4	5	6	7	8	9
Not Very				Very				
Helpful				Helpful				

4. How helpful was the Project Performance Report in your decision making:

1	2	3	4	5	6	7	8	9
Not Very				Very				
Helpful				Helpful				

5. How helpful was the Project Status Report in your decision making:

1	2	3	4	5	6	7	8	9
Not Very				Very				
Helpful				Helpful				

**** PLEASE CONTINUE WITH THE END ****

**** QUESTIONNAIRE ****

QUESTIONS TO BE ANSWERED AFTER COMPLETING ENTIRE EXPERIMENT

1. How clear were the instructions regarding the project?

1	2	3	4	5	6	7	8	9
Not at all Clear								Very Clear

2. How interesting was the task you just performed?

1	2	3	4	5	6	7	8	9
Not at all Interesting								Very Interesting

3. How serious were you in performing the project?

1	2	3	4	5	6	7	8	9
Not at all Serious								Very Serious

4. Have you participated in software project management in the past? _____. If YES, years of experience? _____
Y/N

5. If YES, to what extent was the task in this simulation similar to your previous experience?

1	2	3	4	5	6	7	8	9
Not at all Similar								Very Similar

6. Please give us some information about yourself (in **absolute confidence**. At no time will your name appear in the results. The data will only be used in an aggregate statistical sense).

- (a) Curriculum enrolled in: _____
- (b) Sex _____
- (c) Age _____
- (d) Full time work experience (in years) _____

(e) How long ago (in years) did
you complete your
undergraduate education? _____

(f) How familiar are you with computers, generally?

1	2	3	4	5	6	7	8	9
Not at all								Very
Familiar								Familiar

(g) How many hours (per week) do you use computers?

9. Your general comments regarding the exercise:

END OF EXERCISE

**** THANK YOU FOR YOUR COOPERATION ****

LIST OF REFERENCES

- Abdel-Hamid, T.K., "The Dynamics of Software Project Staffing: A System Dynamics Based simulation approach," *IEEE Transactions on Software Engineering*, v. 15, no.2, pp.109-119. February 1989.
- Boehm, B.W., *Software Engineering Economics*, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1981.
- Bouwman, M.J., "Human Diagnostic Reasoning by Computer: An Illustration from Financial Analysis," *Management Science*, v. 29, no. 6, pp. 653-672, 1983.
- Hogarth, R.M., "Beyond Discrete Biases: Functional and Dysfunctional Aspects of Judgmental Heuristics," *Psychological Bulletin*, v. 90, no. 2, pp. 197-217, 1981.
- Kitfield, J., "Is Software DoD's Achilles' Heel?," *Military Forum*, v. 7, pp. 28-36, 1989.
- Moore, R.M., "Systems and Application Development," *The Economics of Information Processing*, New York, NY: John Wiley & Sons, v. 2, pp. 83-88, 1982.
- Paich, M., and Sterman, J.D., *Boom, bust, and failures to learn in experimental markets*, MIT Sloan School of Management, Working Paper 3441-92-BPS, 1992.
- Payne, J.W., "Task Complexity and Contingent Processing in Decision Making: An Information Search and Protocol Analysis," *Organizational Behavior and Human Performance*, v. 16, pp. 366-387, 1976.
- Remus, W.E., "Graduate Students as Surrogates for Managers in Experiments on Business Decision Making," *Journal of Business Research*, v. 14, pp. 19-25, 1986.
- Remus, W.E., "Testing Bowman's Managerial Coefficient Theory Using a Competitive Gaming Environment," *Management Science*, v. 24, no. 8, pp. 827-835, April 1978.
- Ronan, D., *Decision Making Heuristics and Biases in Software Project Management: An Experimental Investigation*, Master's Thesis, Naval Post Graduate School, Monterey, CA, March 1990.

SAS/STAT, *Guide for Personal Computers*, Version 6, Cary, NC:
SAS, 1987.

Tversky, A., and Kahneman, D., "Judgment Under Uncertainty:
Heuristics and Biases," *Science*, v. 185, pp. 1124-1130,
1974.

Winer, B.J., *Statistical Principles in Experimental Design*,
New York, NY: McGraw-Hill, 1971.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2. Library, code 52 Naval Postgraduate School Monterey, California 93943-5000	2
3. Administrative Sciences Department Naval Postgraduate School Attn: Professor Tung Bui, Code AS/Bd Monterey, California 93943-5000	1
4. Administrative Sciences Department Naval Postgraduate School Attn: Professor Kishore Sengupta, Code AS/Se Monterey, California 93943-5000	1
5. Administrative Sciences Department Naval Postgraduate School Attn: Professor Tarek K. Abdel-Hamid, Code AS/Ah Monterey, California 93943-5000	1
6. LT Timothy P. Costello 1155 Oak Street Red Wing, Minnesota 55066	1

Thesis
C756164 Costello
c.1 Anchoring-and-adjust-
ment in software project
management.

Thesis
C756164 Costello
c.1 Anchoring-and-adjust-
ment in software project
management.

DUDLEY KNOX LIBRARY



3 2768 00034126 7